AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions, and listings, of claims:

Listing of Claims:

1.   (Currently Amended) In a computing environment in which program code is received for one or more objects, a computer-implemented method comprising:

receiving a function call corresponding to the program code in a high level graphics markup language at a visual application programming interface layer in a format which enables the function call to be placed directly to the application program interface layer, wherein the application programming interface layer includes a high-level composition and animation engine which further includes a caching data structure comprising a scene graph of hierarchically-arranged objects;

responding to the function call by causing data in the scene graph to be modified;

invoking a visual manager to traverse the scene graph and render at least one object to a rendering target; and

displaying the scene graph on a display.

2.   (Original) The method of claim 1 wherein causing data in the scene graph to be modified comprises causing initialization of a new instance of a visual class.

3.   (Original) The method of claim 2 wherein causing data in the scene graph to be modified comprises invoking code to associate a transform with a visual object in the scene graph.

4.   (Currently Amended) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking code to place a drawing visual object into the scene graph.

5.   (Currently Amended) The method of claim 4 further comprising, causing a drawing context to be returned, the drawing context providing a mechanism for rendering into the drawing visual object.

6.   (Original) The method of claim 2 wherein causing data in the scene graph to be modified comprises invoking code to associate brush data with a visual object in the scene graph.

7.   (Original) The method of claim 6 wherein the brush data comprises receiving data corresponding to a solid color.

8.   (Original) The method of claim 6 wherein receiving brush data comprises receiving data corresponding to a linear gradient brush and a stop collection comprising at least one stop.

9.   (Original) The method of claim 6 wherein receiving brush data comprises receiving data corresponding to a radial gradient brush.

10.   (Original) The method of claim 6 wherein receiving brush data comprises receiving data corresponding to an image.

11.   (Original) The method of claim 10 further comprising, receiving a function call via an interface corresponding to an image effect to apply to the Image.

12.   (Original) The method of claim 1 further comprising, receiving pen data in association with the function call, and wherein causing data in a scene graph data structure to be modified comprises invoking a pen function that defines an outline of a shape.

13.   (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking code to represent an ellipse in the scene graph data structure.

14.   (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking code to represent a rectangle in the scene graph data structure.

15.   (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking code to represent a path in the scene graph data structure.

16.   (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking code to represent a line in the scene graph data structure.

17.   (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking code related to hit-testing a visual in the scene graph data structure.

18.   (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking code to transform coordinates of a visual in the scene graph data structure.

19.   (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking code to calculate a bounding box of a visual in the scene graph data structure.

20.   (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking code to place a visual object in the scene graph data structure.

21.   (Canceled)

22.   (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking code to place a container object in the scene graph data structure, the container object configured to contain at least one visual object.

23.   (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking code to place image data into the scene graph data structure.

24.   (Original) The method of claim 23 wherein causing data in a scene graph data structure to be modified comprises invoking code to place an image effect object into the scene graph data structure that is associated with the image data.

25.   (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking code to place data corresponding to text into the scene graph data structure.

26.   (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking code to provide a drawing context in response to the function call.

27.   (Original) The method of claim 26 wherein the function call corresponds to a retained visual, and further comprising, calling back to have the drawing context of the retained visual returned to the scene graph data structure.

28.   (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking code to place a three-dimensional visual into the scene graph data structure.

29.   (Original) The method of claim 28 wherein causing data in a scene graph data structure to be modified comprises invoking code to map a two-dimensional surface onto the three dimensional visual.

30.   (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking code to place animation data into the scene graph data structure.

31.   (Original) The method of claim 30 further comprising communicating timeline information corresponding to the animation data to a composition engine.

32.   (Original) The method of claim 31 wherein the composition engine interpolates graphics data based on the timeline to animate an output corresponding to an object in the scene graph data structure.

33.   (Original) The method of claim 1 wherein receiving a function call via an interface comprises receiving markup, and wherein causing data in a scene graph data structure to be modified comprises parsing the markup into a call to an interface of an object.

34.   (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking code to place an object corresponding to audio and/or video data into the scene graph data structure.

35.   (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking code to change a mutable value of an object in the scene graph data structure.

36-64   (Canceled)

65.   (Previously Presented)   The method recited in claim 1, wherein the method further includes the high-level composition and animation engine passing instructions to a low-level compositing and animation engine and which is thereafter sent to a graphics subsystem.

66.   (New) In a computing environment in which program code is received for one or more objects, a computer-implemented method comprising:

receiving a function call corresponding to the program code in a high level graphics markup language at a visual application programming interface layer in a format which enables the function call to be placed directly to the application program interface layer, wherein the application programming interface layer includes a high-level composition and animation engine which further includes a caching data structure comprising a scene graph of hierarchically-arranged objects;

responding to the function call by causing initialization of a new instance of a visual class and invoking code to associate data corresponding to an image with a visual object in the scene graph, thereby causing data in the scene graph to be modified;

receiving a function call via an interface corresponding to an image effect to apply to the image;

invoking a visual manager to traverse the scene graph and render at least one object to a rendering target; and

displaying the scene graph on a display.